Proposal for selecting a cooperation partner in distributed control of traffic signals using deep reinforcement learning

Shinya Matsuta ^{a,*}, Naoki Kodama ^b, Taku Harada ^a

^a Tokyo University of Science, 2641 Yamazaki, Noda, Chiba, 278-8510, Japan

^b Meiji University, 1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

*Corresponding Author: 7420527@ed.tus.ac.jp

Abstract

Traffic signal control is one way to alleviate traffic congestion on road networks. The main method of traffic signal control is a distributed control method in which signals cooperate locally. In this study, to realize more effective control in the distributed control system, we propose a guideline for selecting the cooperation partner of each traffic signal and verify its effectiveness. In this study, the traffic signal is controlled by applying deep reinforcement learning, which is a machine-learning algorithm.

Keywords: traffic signal control, cooperation, distributed control, deep reinforcement learning, reward, agent.

1. Introduction

One way to alleviate frequently occurring traffic congestion is to control the traffic signals. In various existing studies, reinforcement learning, which is a machine learning algorithm, has been applied for traffic signal control⁽¹⁾.

There are two main methods of traffic signal control: centralized control and distributed control. Centralized control collectively controls all traffic signals existing in the entire target road network at the control center. However, the amount of calculation required for control increases significantly as the number of intersections to be handled increases. Furthermore, the equipment failure at a control center can put the entire target road network in risk.

In distributed control, the traffic signals cooperate locally to obtain the optimum phase of each traffic signal.

This study focuses on the distributed traffic signal control method. While using distributed control, it is often the case that the cooperation partner comprises all the adjacent traffic signals. However, as there are often

differences in traffic volume on each road, it is possible to learn more effective control by limiting the cooperation partners than by cooperating with all adjacent traffic signals.

We have been studying the cooperation control of traffic signals using deep reinforcement learning⁽²⁾. Our previous study⁽²⁾ shows the effectiveness of cooperating only with the adjacent traffic light with the largest inflow traffic in the mesh-shaped road network.

In this study, we first introduce the method proposed in our study⁽²⁾. Furthermore, we propose a method for selecting a cooperation partner more flexibly and verify its effectiveness. In this study, it is not a direct cooperation method, such as sharing the indication state of the adjacent traffic signal, but indirectly adjacent traffic signals, by recognizing the traffic condition on the road flowing into oneself.

In this study, the control of traffic signals is learned using Deep Q-Network (DQN)⁽³⁾, a deep reinforcement learning method. The shape of the road network used in this study is a mesh.

2. Related works

Hongwei et al.⁽⁴⁾ proposed QT-CDQN (cooperative deep Q-network with Q-value transfer) for cooperated traffic signal control. This includes not only the Q value of one's own traffic signal, but also the Q value of an adjacent traffic signal. Misawa et al.⁽⁵⁾ performed cooperation control in an environment where the traffic volume on all routes was not constant. Therefore, the ratio of straight ahead at intersections increases. Suzuki et al.⁽⁶⁾ alleviated traffic congestion at intersections when the traffic volume was equal on each road, considering that the degree of time loss and economic loss differ depending on the vehicle type. Diamantis et al.⁽⁷⁾ performed adaptive fine-tuning (AFT)

with real traffic and demonstrated its effectiveness. Tian et al.⁽⁸⁾ applied Coder, a deep reinforcement learning algorithm, for traffic signal control. Min et al.⁽⁹⁾ applied the A2C algorithm for traffic signal control and performed it on an existing road network. Daeho et al.⁽¹⁰⁾ predicted future traffic conditions and performed a cooperation control. Deepak et al.⁽¹¹⁾ proposed cooperative multi-agent reinforcement learning-based models for traffic control optimization.

However, a large number of existing studies focus on cooperating with all adjacent traffic signals. Few studies have shown the effectiveness of selecting partners.

3. Deep Q-Network

Reinforcement learning seeks a policy to maximize the rewards obtained by a learner (agent) who acts on the environment. In reinforcement learning, the optimal policy is obtained by repeating the following three steps: (1) the agent takes action, (2) the state of the environment is updated by the action of the agent, and in some cases, a reward is obtained for the action; and (3) the agent modifies the policy based on the reward. Reward is an index showing the goodness of an agent's action and environmental state. A policy is a rule that serves as an index when an agent acts.

In Q-learning⁽¹²⁾, the value of an agent's action is managed in the Q table, and the Q value is updated each time the action is taken. The Q value indicates a value that takes into account future rewards. Deep Q-Network (DQN)⁽³⁾ extends Q-learning by including neural networks in Q-learning. Specifically, a neural network is used to approximate the Q value of each action with respect to the state of the environment. Then, if the Q value for each action can be estimated in a certain state, the action with the best Q value, that is, the best action to be taken, can be known. The Q value is calculated as the expected value of the discounted income of the rewards that can be obtained in the future. The updated formula for the Q value is given by Eq. (1):

$$Q(s, a, \theta) \leftarrow R(s, a) + \gamma \max_{a'} Q(s', a', \theta) \quad (1)$$

Where Q is the action value, s is a state, γ is the current value of a future reward, a is an action, R is a reward, and θ is a learning parameter of a neural network. Experience replay and target network methods are used in the DQN to stabilize learning. Experience replay accumulates past experiences in an array called replay memory, selects from the accumulated experiences randomly, and updates the Q value as supervised data. The target network separates the parameters of the Q function of the

target and the Q function to be updated. This stabilizes the learning by fixing the target for some time.

It is difficult for Q-learning to express continuous values. However, because DQN is based on the idea of obtaining an approximate function of the Q value by using a neural network, it is possible to deal with continuous values.

4. Applying DQN

We describe the cooperation method proposed in our previous study⁽²⁾. We then propose a method for selecting a cooperation partner more flexibly. The "action selection", "state recognition", and "reward design" are the same as in Reference⁽²⁾.

4.1 Agent

The traffic signal at each intersection is treated as an agent. Each traffic signal agent selects an action after perceiving the state and receives a reward from the environment depending on the result of the action performed. One cycle is defined as a step.

4.2 Action selection

The actions that the traffic signal agent can select are two types of phases: "north-south is red light + east-west is green light" and "north-south is green light + east-west is red light". When switching phases, the yellow signal phase is inserted regardless of the intention of the traffic signal agent.

4.3 State recognition

The traffic signal agent recognizes five items: the traffic signal phase within one step range, the duration of the phase [s], the traffic density [vehicles/km], the number of waiting vehicles [units], and the average speed [km/h]. The traffic signal itself has two phase patterns: "green light for north and south + red light for east and west" and "red light for north and south + green light for east and west". The duration of the phase indicates the time during which "north-south is green light + east-west is red light" or "north-south is red light + east-west is green light". The traffic density, number of waiting vehicles, and average speed indicate the density of roads, the number of vehicles stopped, and the average speed in each of the four directions. The recognition ranges are shown in Figure 1.

4.4 Reward design

Reward R was set using v' , which is normalized to a value of approximately 0 to 1 for an average speed v of all

Fig. 1. Range for state recognition

vehicles existing within the recognition range. Regarding normalization, because the velocity can take a value slightly over 100 [km/h] from 0 [km/h], the observed velocity was divided by 120 to take a range of approximately 0 to 1.

$$R = v' \quad (2)$$

$$v' = \frac{v}{120} \quad (3)$$

4.5 Selection of cooperation partner

The degree of variation in traffic volume throughout the road network varies. In other words, the degree of variation in the amount of traffic flowing into each intersection varies. However, as in our previous study⁽²⁾, fixing only the adjacent intersections of the lanes, which has the largest inflow, as a cooperation partner does not take this variation into consideration. Therefore, to select the cooperation partner more flexibly, the selection is made based on the ratio of the traffic volume flowing into a traffic signal. First, we find the ratio of traffic volume flowing in from the four directions. Then, the traffic signal cooperates only with the traffic signal of the lane of the traffic volume exceeding the preset threshold value. If there are no lanes with traffic exceeding the threshold, it cooperates with traffic signals in all four directions.

5. Evaluation experiment

5.1 Experimental environment

An open-source micro traffic flow simulator, Simulation of Urban MObility (SUMO)⁽¹³⁾, which provides

API and GUI, was used as the simulation environment.

As shown in Fig. 2, the shape of the road network is a 4×4 mesh, and the length of one side is 400 [m]. Therefore, the length of one continuous road was 2 [km].

To congest some routes, vehicles appeared with a probability of 0.9, in places with heavy traffic. Furthermore, the number of vehicles that appeared with a probability of 0.9 was set to appear up to 100 in each step. Other vehicles appear with a probability of 0.0011 for each step. All vehicles emerge from the edge and disappear when they reach the edge.

When the vehicle travels to the destination, the shortest route is followed to reach the destination. Therefore, it is possible to detour and arrive at the destination.

In Figure 2, A to F indicate columns, and 0 to 5 indicate rows. Each intersection is written as A1, A2, A3, using row and column symbols, respectively. The arrows indicate roads with heavy traffic.

The following three roads were set as roads with heavy traffic. (1) The road that first passes through the traffic signal at intersection B4 and passes through intersections C4, C3, C2, and C1. (2) The road that first passes through the traffic signal at intersection B4 and passes through intersections C4, C3, D3, and E3. (3) The road that first passes through the traffic light at intersection B4 and passes through intersections C4, C3, D3, D2, and D1. In addition, 90 [%] of the vehicles passing through traffic light D3 proceed to E3, and 10 [%] proceed to D2 and pass through D1.

The road is intended as a general road, but considering its application in the international road network, the speed limit of general roads is considered to be up to 90 [km/h].

At each intersection, the vehicle is set so that it can go straight, turn left, turn right, and make a U-turn.

Agents are set for all traffic signals, resulting in a total of 16 traffic signal agents. The maximum time that a traffic signal can continue in the red and green phases is 50 [s], and the minimum time is 5 [s]. When switching phases, a yellow signal phase of 2 [s] is always inserted regardless of the intention of the traffic signal agent.

5.2 Experimental scenario

Two experimental scenarios were set.

[Scenario 1]

Set 30 [%] as the threshold.

[Scenario 2]

All traffic signal agents recognize the states of all four adjacent locations. That is, the traffic signal agents in cooperates in four directions.

Fig. 2. Road network

5.3 Parameter setting

The neural network that calculates the Q value has three hidden layers consisting of 256 nodes, with the nodes between each layer are fully connected. The input is four-dimensional, whereas the output is two-dimensional. We used an Adam optimizer⁽¹⁴⁾ as a gradient descent method to minimize the error of the Q value, with the parameters following Adam's Reference⁽¹⁴⁾. In addition, the ϵ -greedy method was adopted for the selection of actions. ϵ was linearly reduced to 0.02 after 10000 steps from the initial value. The remaining parameters follow the DQN Reference⁽³⁾.

One cycle is defined as a step, and the action in one step is assumed to be executed continuously for 5 [s].

The simulation time was set to 10000 [s]. We waited for 500 [s] before starting learning so as to let enough vehicles enter the road network. This procedure was repeated 1000 times.

5.4 Experimental results and discussion

Two executions of scenario 1 and scenario 2 were executed 10 [times] each.

Table 1 shows the cooperation partners in one of the ten executions in Scenario 1.

Table 2 shows the execution result of scenario 1, and Table 3 shows the execution result of scenario 2. These two tables show the average speed and reward for the final result of the run. Figure 3 shows the average curve at the average velocities of scenario 1 and scenario 2. The average speed is the average speed of the entire road network in 10 executions. The vertical axis shows the average speed [km/h], and the horizontal axis shows the number of learnings. The blue line shows the learning results in Scenario 1, and the orange line shows the learning results in Scenario 2. As shown in Table 2, Table 3, and Figure 3, the learning result in Scenario 1 is superior. Furthermore, it can be seen that in the early stages

Table 1. Cooperation destination of each intersection

Intersection	Ratio	Cooperation destination
B1	{'A1': 0.23, 'B0': 0.24, 'B2': 0.30, 'C1': 0.23}	B2
B2	{'A2': 0.23, 'B1': 0.25, 'B3': 0.30, 'C2': 0.21}	B3
B3	{'A3': 0.24, 'B2': 0.26, 'B4': 0.27, 'C3': 0.23}	A3,B2,B4,C3
B4	{'A4': 0.96, 'B3': 0.01, 'B5': 0.01, 'C4': 0.02}	A4
C1	{'B1': 0.07, 'C0': 0.04, 'C2': 0.81, 'D1': 0.08}	C2
C2	{'B2': 0.05, 'C1': 0.03, 'C3': 0.87, 'D2': 0.05}	C3
C3	{'B3': 0.02, 'C2': 0.02, 'C4': 0.94, 'D3': 0.02}	C4
C4	{'B4': 0.97, 'C3': 0.01, 'C5': 0.01, 'D4': 0.01}	B4
D1	{'C1': 0.24, 'D0': 0.22, 'D2': 0.26, 'E1': 0.28}	C1,D0,D3,E2
D2	{'C2': 0.24, 'D1': 0.24, 'D3': 0.27, 'E2': 0.25}	C2,D1,D3,E3
D3	{'C3': 0.39, 'D2': 0.20, 'D4': 0.22, 'E3': 0.19}	C3
D4	{'C4': 0.24, 'D3': 0.26, 'D5': 0.24, 'E4': 0.26}	C4,D3,D5,E4
E1	{'D1': 0.25, 'E0': 0.22, 'E2': 0.27, 'F1': 0.26}	D1,E0,E2,F1
E2	{'D2': 0.22, 'E1': 0.29, 'E3': 0.25, 'F2': 0.25}	D2,E1,E3,F2
E3	{'D3': 0.34, 'E2': 0.27, 'E4': 0.23, 'F3': 0.16}	D3
E4	{'D4': 0.26, 'E3': 0.28, 'E5': 0.24, 'F4': 0.23}	D4,E3,E5,F4

Table 2. Execution results of scenario 1

Time	Average speed[km/h]	Reward
1 st	92.42	0.7701
2 nd	92.15	0.7679
3 rd	91.07	0.7589
4 th	91.83	0.7655
5 th	91.60	0.7634
6 th	91.66	0.7639
7 th	93.69	0.7808
8 th	92.34	0.7696
9 th	91.87	0.7656
10 th	92.88	0.774
Average	92.15	0.768

Table 3. Execution results of scenario 2

Time	Average speed[km/h]	Reward
1 st	90.63	0.7136
2 nd	89.24	0.7404
3 rd	90.61	0.7563
4 th	89.71	0.7526
5 th	91.06	0.7645
6 th	90.32	0.7593
7 th	91.3	0.7649
8 th	90.67	0.7522
9 th	90.69	0.7634
10 th	90.7	0.7626
Average	90.48	0.754

Fig. 3. Average speed

of learning, Scenario 1 learns faster than Scenario 2. From these experimental results, it can be seen that it is important not to cooperate in consideration of the conditions of all adjacent roads, but to appropriately select the cooperation partner.

In this experiment, the average speed of the vehicle could be improved by selecting a cooperation partner. This can be attributed to the increase in the average speed of roads with heavy traffic by recognizing the state of only the road with heavy traffic. Congestion on high-traffic roads has a greater impact on the entire road network than on low-traffic roads. Therefore, the average speed of the entire network is increased by smoothing the traffic flow on the road with heavy traffic.

6. Conclusion

In this study, we proposed a method for selecting a cooperation partner in distributed control of traffic signals using deep reinforcement learning. The proposed method cooperated only with the adjacent traffic signals in the lanes with traffic volumes whose ratio to the total traffic flowing into the traffic signals exceeded a preset threshold. To evaluate the effectiveness of the proposed method, a simulation experiment was conducted in a 4×4 signal network with increasing traffic volume on a specific road. The experimental results revealed that the proposed method improved the average speed of the road network by approximately 1.7 [%]. This result can be attributed to effective traffic control on a busy road. Thus, the effectiveness of our proposed method was verified.

Although a threshold value was set in the proposed method, a method for setting dynamic threshold value depending on the traffic volume should be established in

future work.

References

- (1) Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, Peter Komisarczuk: "A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control", *ACM Computing Surveys*, Vol. 50, Issue 3, (2017)
- (2) Shinya Matsuta, Naoki Kodama, Taku Harada: "Basic Study on Distributed Traffic Signals Control", *Technical Meeting on Systems, The Institute of Electrical Engineers of Japan*, pp.5-8, (2021) (in Japanese)
- (3) Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Alex. Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, Demis Hassabis: "Human-level control through deep reinforcement learning", *NATURE*, vol.518, pp.529-541, (2015)
- (4) Hongwei Ge, Yumei Song, Chunguo Wus, Jiankang Ren, Guozhen Tan: "Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control", *IEEE Access*, pp.40797-40809, (2019)
- (5) Tadanobu Misawa, Haruhiko Kimura, Hirose Sadaki, Nobuyasu Osato: "Multiagent-based Traffic Signal Control with Reinforcement Learning", *The Institute of Electronics Information and Communication Engineers Transactions on Information and Systems*, Vol. J83-D-I, No.5, pp. 478-486, (2000) (in Japanese)
- (6) Ryoji Suzuki, Katsuhide Fujita: "Reinforcement learning for multiple rewards to alleviate traffic congestion at intersections", *2020 Information Processing Society of Japan*, pp.75-76, (2020) (in Japanese)
- (7) Diamantis Manolis, Ioannis Papamichail, Elias Kosmatopoulos, Markos Papageorgiou: "Automated tuning of ITS management and control systems: Results from real-life experiments", *Transportation Research Part C: Emerging Technologies* Vol. 66, pp.119-135, (2016)
- (8) Tian Tan, Feng Bao, Yue Deng, Alex Jin, Qionghai Dai, Jie Wang: "Cooperative Deep Reinforcement Learning for Large-Scale Traffic Grid Signal Control", *IEEE Transactions on Cybernetics*, Vol. 50, No. 6, (2020)

- (9) Min Wang, Libing Wu, Jianxin Li, Liu He: "Traffic Signal Control With Reinforcement Learning Based on Region-Aware Cooperative Strategy", IEEE Transactions on Intelligent Transportation Systems, pp.1-12, (2021)
- (10) Daeho Kim, Okran Jeong: "Cooperative Traffic Signal Control with Traffic Flow Prediction in Multi-Intersection", Creative Commons Attribution, (2019)
- (11) Deepak Vidhate, Parag Kulkarni: "Cooperative Multi-agent Reinforcement Learning Models (CMRLM) for Intelligent Traffic Control", Proceeding of 2017 1st International Conference on Intelligent Systems and Information Management, pp.325-331, (2017)
- (12) Chris Jch Watkins, Peter Dayan: "Technical note: Q-learning", Machine Learning, 8, pp.279-292, (1992)
- (13) Daniel Krajzewicz, Georg Hertkorn, Christian Feld, Peter Wagner: "SUMO (Simulation of Urban MObility); An open-source traffic simulation", 4th Middle East Symposium on Simulation and Modelling, (2002)
- (14) Diederik Kingma, Jimmy Ba: "Adam: A Method for Stochastic Optimization", Proceeding of The International Conference on Learning Representations 2015, (2015)